ITERATIVE METHODS FOR LINEAR SYSTEMS AND IMPLEMENTATION ON PARALLEL COMPUTERS

Henk Van der Vorst

Utrecht University

OUTLINE

- Motivation
- Basic ideas
- CG, GMRES, BICG, QMR
- Composite Methods:
CGS, Bi-CGSTAB(ℓ), GMRESR
- Numerical aspects
- Performance aspects

WHY ITERATIVE METHODS?

When direct methods too expensive

When multigrid not feasable

— iterative methods may do —

ADVANTAGE OF ITERATIVE METHODS:

modest in time per step

modest in memory requirements

DISADVANTAGE: not robust

EXAMPLE

- PDE in 3D over irregular grid with $\approx m$ unknowns in each direction

$Ax = b$, $n = m^3$, bandwidth $m^2 = n^{\frac{2}{3}}$

DIRECT METHOD (Gauss. elim.)

work $\sim \mathcal{O}(n^{2\frac{1}{3}})$, storage $\sim nm^2$

ITERATIVE METHOD (Conj. Grad.)

error reduction $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$

gridproblems: $\kappa \sim m^2 = n^{\frac{2}{3}}$

work per step fn, reduction ϵ:

$\Longrightarrow$ work for CG $\sim -\log \epsilon \cdot fn^{\frac{4}{3}}$

ILLUSTRATION

Finite element space model (Simon '89)

Realistic model: $n = 5 \star 10^9$

Direct Solver at 1 TFLOP: $520,040$ years

(exploitable parallelism)

Conjugate Gradients: $\sim$ 500 iteration steps

With 1 TFLOP speed: 30 minutes

(parallelism exploitable?)

ITERATIVE METHODS - BACKGROUND

$$Ax = b, \quad \text{write:} \quad A = I - (I - A)$$

$$(I - (I - A))\, x = b$$

$$\Downarrow$$

BASIC ITERATION METHOD:

$$\begin{aligned} x_i &= b + (I - A)x_{i-1} \\ &= x_{i-1} + r_{i-1} \\ &= x_{i-2} + r_{i-2} + r_{i-1} \end{aligned}$$

BASIC ITERATION METHOD

$$x_i = x_0 + r_0 + r_1 + \cdots + r_{i-1}$$

from $x_i = x_{i-1} + r_{i-1}$

$A\times \;\Rightarrow\; Ax_i = Ax_{i-1} + Ar_{i-1}$

$$\begin{array}{rcccl} b - & \Rightarrow & \underbrace{b - Ax_i}_{r_i} & = & \underbrace{b - Ax_{i-1}}_{r_{i-1}} - Ar_{i-1} \\ & & & = & r_{i-1} - Ar_{i-1} \\ & & & = & (I - A)r_{i-1} \end{array}$$

$$\begin{aligned} x_i &= x_0 + r_0 + (I - A)r_0 + .. + (I - A)^{i-1}r_0 \\ &= x_0 + \left[r_0, Ar_0, A^2r_0, \ldots, A^{i-1}r_0\right] \end{aligned}$$

KRYLOV SUBSPACE METHODS

$$x_i = x_0 + \left[r_0, Ar_0, A^2 r_0, \ldots, A^{i-1} r_0\right]$$

$\Uparrow$ **KRYLOV subspace**

CAN WE DO BETTER?

Select vector from Krylov subspace
such that $\|Ax_i - b\|_2$ is minimal :
GMRES (MINRES)

or $Ax_i - b \perp$ Krylov subspace:
FOM (CG), Bi-CG

ORTHOGONALITY OF RESIDUAL

find $x_i \in K^i(A; r_0)$ such that

$$r_i \equiv b - Ax_i \perp K^i(A; r_0)$$

Take $x_0 = 0$: $r_0 = b$

Then $x_1 = \mu r_0$ and $r_1 = r_0 - \mu A r_0 \in K^2(A; r_0)$

Since $r_1 \perp r_0 \quad \Rightarrow \quad \{r_0, r_1\}$ form an orthogonal basis for $K^2(A; r_0)$

INDUCTION: $r_0, r_1, \ldots, r_{i-1}$ form an orthogonal basis for $K^i(A; r_0)$

CONSTRUCTION OF BASIS

suppose $r_0, \ldots, r_i$ orthogonal
and A is symmetric

$$\gamma_i r_{i+1} = Ar_i - \alpha_i r_i - \beta_i r_{i-1} - \delta_i r_{i-2} - \cdots$$

$$(r_{i+1}, r_{i-2}) = 0 \quad \Rightarrow$$

$$\delta_i(r_{i-2}, r_{i-2} = (Ar_i, r_{i-2}) =$$

$$(r_i, Ar_{i-2}) = (r_i, \gamma_{i-2} r_{i-1} + \alpha_{i-2} r_{i-2} + \cdots) = 0$$

3-TERM RECURRENCE

$$\gamma_i r_{i+1} = Ar_i - \alpha_i r_i - \beta_i r_{i-1}$$

COEFFICIENTS

$$\gamma_i r_{i+1} = A r_i - \alpha_i r_i - \beta_i r_{i-1}$$

$$(r_{i+1}, r_i) = 0 \quad \Rightarrow \quad \alpha_i = (A r_i, r_i)/(r_i, r_i)$$

$$(r_{i+1}, r_{i-1}) = 0 \;\Rightarrow\; \beta_i = (A r_i, r_i)/(r_{i-1}, r_{i-1})$$

Polynomial Expression $x_i = Q_{i-1}(A) r_0$

HENCE

$r_i = b - A x_i = (I - A Q_{i-1}(A)) r_0 = P_i(A) r_0$

Note that $P_i(0) = 1$

Insert in recurrence, coefficient for r_0:

$$\gamma_i = -\alpha_i - \beta_i$$

MATRIX REPRESENTATION

$$\gamma_i r_{i+1} = Ar_i - \alpha_i r_i - \beta_i r_{i-1}$$

R_i: matrix with columns $r_0, \ldots, r_i$

THEN

$$AR_i = R_i T_i + \begin{pmatrix} 0 & 0 & & 0 & \\ 0 & 0 & & 0 & \\ \vdots & \vdots & & \vdots & \gamma_i r_{i+1} \\ 0 & 0 & & 0 & \\ 0 & 0 & & 0 & \end{pmatrix}$$

$$T_i = \begin{pmatrix} \ddots & \ddots & & & & & \\ \ddots & \ddots & \ddots & & & & \\ & \ddots & \ddots & \beta_j & & & \\ & & \ddots & \alpha_j & \ddots & & \\ & & & \gamma_j & \ddots & \ddots & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \ddots & \ddots \end{pmatrix}$$

REDUCED SIZE SYSTEM

$$r_{i+1} = b - Ax_{i+1} \perp \{r_0, \ldots, r_i\}$$

OR $\quad R^T(b - Ax_{i+1}) = 0$

AND

$x_{i+1} \in \{r_0, \ldots, r_i\} \quad \Rightarrow \quad x_{i+1} = Ry_{i+1}$

HENCE $R^T ARy_{i+1} = R^T b$

WITH $AR = RT + \gamma_{i+1} r_{i+1} e_{i+1}^T$

$$R^T RTy_{i+1} = R^T b = R^T r_0 = \|r_0\|^2 e_1$$

REDUCED SYSTEM:

$Ty_{i+1} = e_1$ with $x_{i+1} = Ry_{i+1}$

CONJUGATE GRADIENTS

$Ax = b$ with basis $\{r_0, \ldots, r_i\}$ $\quad \Rightarrow \quad$ $Ty_{i+1} = e_1$

if A is *SYMMETRIC POSITIVE DEFINITE* then

T is *symmetric positive definite*

$\Downarrow$

T can be split $\Rightarrow$ simple recursions

$\Downarrow$

CG

CONJUGATE GRADIENTS Hestenes & Stiefel '52

KRYLOV SUBSPACE METHODS (2)

Create orthonormal basis for
$\left[r_0, Ar_0, \ldots, A^{m-1}r_0\right]$:
$v_1 = r_0, v_2, \ldots, v_m$

From v_i: compute Av_i,
orthogonalise w.r.t. v_1,..,v_i

$h_{i+1,i}v_{i+1} = Av_i - \sum_1^i h_{j,i}v_j$

$v_{i+1} \perp v_j$: $h_{j,i} = (Av_i, v_j)$
$\|v_{i+1}\|_2 = 1$: $h_{i+1,i}$

Result: $AV_m = V_{m+1}H_{m+1,m}$
(V_m has columns v_1, ..., v_m)

KRYLOV SUBSPACE METHODS (3)

$x_m \in$ Krylov subspace: $x_m = V_m y$

(1) Galerkin approach
require $r_m = b - Ax_m \perp$ Krylov subspace
(r_i: orthogonal basis):

$0 = V_m^T r_m$: $V_m^T A x_m = V_m^T b$

$\Rightarrow$: $V_m^T A V_m y = \tilde{b}$
with $AV_m = V_{m+1} H_{m+1,m}$:

$$H_{m,m} y = \tilde{b}$$

This leads to the FOM method
A symm.: $H_{m,m}$ tridiag.: LANCZOS
A pos.def.: CG

KRYLOV SUBSPACE METHODS (4)

$x_m \in$ Krylov subspace: $x_m = V_m y$

(2) Minimum norm approach:
Minimise $\|b - AV_m y\|_2$:

$\|b - AV_m y\| = \|b - V_{m+1} H_{m+1,m} y\|$

y minimizes $\|H_{m+1,m} y - V_{m+1}^T b\|$

This leads to GMRES
A symm.: $H_{m+1,m}$ tridiag.: MINRES

SKETCH OF GMRES

FOR $j = 1, 2, \ldots, m$

$\quad w = Av_j$

$\quad$ FOR $i = 1, \ldots, j$

$\quad\quad h_{ij} = (w, v_i)$

$\quad\quad w = w - h_{ij} v_i$

$\quad h_{j+1,j} = \|w\|_2$; $v_{j+1} = w / \|w\|_2$

Solve $H_m y_m = \|r_0\| e_1$

$x = x_0 + [v_1, \ldots, v_m]\, y_m$

if necessary repeat

GMRES

GMRES (=GCR, ORTHODIR, ORTHOMIN)

- optimal w.r.t. MV's:

m steps; m MV's

BUT:

computational costs $\sim m^2$
memory space $\sim m$

- Cheaper processes?

- How to keep m small?

NONSYMMETRIC 3-TERM

When A is nonsymmetric $\Rightarrow$
NO 3-TERM ORTHOGONAL BASIS

HOWEVER: BI-ORTHOGONAL BASES :

$\{r_0, \ldots, r_i\}$ FOR $\{r_0, Ar_0, \ldots, A^i r_0\}$

AND

$\{s_0, \ldots, s_i\}$ FOR $\{s_0, A^T s_0, \ldots, {A^T}^i s_0\}$

WITH

$$(r_k, s_j) = 0 \text{ for } k \neq j$$

3-term recursion for $\{r_k\}$ (with A)
and one for $\{s_j\}$ (with A^T)

Bi-CONJUGATE GRADIENTS

MATRIX FORMULATION:

$AR = RT + \gamma_{i+1} r_{i+1} e_{i+1}^T$

$x_{i+1} = Ry_{i+1}$ and $S^T(b - Ax_{i+1}) = 0$

$$\Downarrow$$

$$S^T A R y_{i+1} = S^T b$$

OR

$$S^T RTyy_{i+1} = S^T b \text{ with } x_{i+1} = Ry_{i+1}$$

$$\Downarrow$$

BI-CG

Lanczos '52, Fletcher '76

Bi-Conjugate Gradients: BiCG

With 3-term recursion:
construction of basis $\{v_j\}$ with
$v_j \perp \left[v_1, A^T v_1, ..., (A^T)^{j-1} v_1\right]$

Bi-CG:
x_m such that $r_m \perp K^m(A^T; r_0)$ (basis)

- not optimal in $K^i(A; r_0)$
- 2 MV's per iteration
- CG-like costs
- more iterations than GMRES:
  $\|Ax_i^{BiCG} - b\|_2 \geq \|Ax_i^{GMRES} - b\|_2$

LEAST SQUARES APPROACH

$$\left.\begin{array}{c} Ax = b \;\;\Rightarrow\;\; A^TAx = A^Tb \\ + \\ CG \end{array}\right\} \Rightarrow \text{LSQR}$$

stable implementation: Paige & Saunders '82

disadvantage: squared condition number

suitable for overdetermined systems

OVERVIEW OF METHODS

A sym. pos. def. :

CONJUGATE GRADIENTS

A symmetric :

LANCZOS, MINRES

A unsymmetric:

- **LSQR** { CGLS, CRAIG, ...}
- **GMRES** { FOM, ORTHOMIN, GMRESR, ORTHODIR, ...}
- **BiCG** { QMR, BiCGSTAB, TFQMR, CGS, ...}

TFQMR, CGS, ...}

VARIANTS OF BiCG (1)

• **QMR** (Freund & Nachtigal)
- slighty better than Bi-CG
- more smooth convergence
- more iterations than GMRES

• **CGS** (Sonneveld)
- 2 MV's in BiCG can be used to apply BiCG twice: CGS
- same costs as BiCG
- often twice as fast
- very irregular convergence
- often faster than GMRES
- more MV's than GMRES

VARIANTS OF Bi-CG (2)

Basis for $K^m(A; r_0)$ and $K^m(A^T; r_0)$
with same 3-term recursions:

$$r_i = P_i(A)r_0$$

$$\text{but also: } \widehat{r}_i = P_i(A^T)r_0$$

Bi-CG coëfficients through
innerproducts like $(Ar_i, \widehat{r}_j)$

Sonneveld (1984):

$$(Ar_i, \widehat{r}_j) = (AP_i(A)r_0, P_j(A^T)r_0)$$

$$= (AP_i(A)P_j(A)r_0, r_0)$$

$\widehat{r}_j$ not necessary!!
Simple recursion for $\widetilde{r}_i = P_i^2(A)r_0$: CGS

ARE PEAKS BAD?

Many iteration processes:

$$\begin{aligned} x_i &= x_{i-1} + \alpha_i p_i \\ r_i &= r_{i-1} - \alpha_i A p_i \end{aligned}$$

errors in x_i no effect on r_i

In finite precision:

$$r_i = r_{i-1} - \alpha_i A p_i - \alpha_i \Delta_A p_i$$

$$|\Delta_A| \leq n_A \xi |A|$$

$$r_i - (b - Ax_i) = -\sum_{j=1}^{i} \alpha_j \Delta_A p_j$$

$$|\|r_i\|_2 - \|b - Ax_i\|_2| \leq$$

$$2\, i\, n_A \xi \||A|\| \|A^{-1}\| \max_j \|r_j\|$$

RELIABLE UPDATING

from suggestion by Neumaier '94
made for CGS

$\bar{x} = x_0$; $\bar{r} = r_0$; $x_u = 0$

...

for $i = 0, 1, 2, \ldots$

$\quad$ ...

$\quad x_u = x_u + \alpha_i p_i$

$\quad r = r - \alpha_i A p_i$

$\quad$ ...

$\quad$ **if** $\|r\| < \|\bar{r}\|$

$\qquad \bar{x} = \bar{x} + x_u$

$\qquad \bar{r} = r = b - A\bar{x}$

$\qquad x_u = o$

$\quad$ **endif**

if $\|r\| \approx \xi \|r_0\|$: $r_i \approx b - Ax_i$

for further info: Sleijpen and VDV '94

MOTIVATION TO IMPROVE CGS

GOALS:

- smoother convergence
- faster convergence
- maintain quadratic effects
  in converged eigendirections

POSSIBILITIES:

- different s_0
- instead of $r_i = P_i^2(A) r_0$:

$$r_i = \widetilde{P}_i(A) P_i(A) r_0$$

with "nearby" $\widetilde{P}_i$

NEARBY POLYNOMIALS

$$r_i = \widetilde{P}_i(A) P_i(A) r_0$$

(1) $P_i(A)$ is based on r_0 and s_0

Make $\widetilde{P}_i(A)$ with r_0 and $t_0 \neq s_0$

Often for $\widetilde{P}_i$ and P_i

$$\widetilde{P}_i(\lambda_j) \approx P_i(\lambda_j) \approx 0$$

for some λ_j and i not too small
quadratic convergence in some directions

NEARBY POLYNOMIALS (2)

Take $\widetilde{P}_i(A) = (I - \omega A)P_{i-1}(A)$
Often much smoother behavior

Heuristic explanation:
If A is symmetric then zeros of P_i
separate those of P_{i-1}
$\Rightarrow$ zeros of P_{i-1} kill peaks of P_i

For converged zeros: $P_i(\lambda_j) \approx P_{i-1}(\lambda_j) \approx 0$
again quadratic effects

joint work with Fokkema & Sleijpen, 1995

VARIANTS OF BiCG (3)

- **BiCGSTAB** (Van Der Vorst)

- 2 MV's in BiCG can be used to combine
  BiCG with repeated GMRES(1)
- same costs as BiCG
- often much faster than BiCG
- much smoother than CGS
- often faster than CGS
- breakdown when GMRES(1) stagnates
- poor when GMRES(1) is very poor

VARIANTS OF BiCG (4)

- **BiCGSTAB(2)**

  (Sleijpen, Fokkema, vdVorst '94)

- after each two BiCG steps: GMRES(2)

- often faster than BiCGSTAB

- also for near skew-symm. matrices

- $\approx$ same costs as BiCG

- easy to generalize: BiCGSTAB(ℓ)

- BiCGSTAB(4): fast and robust

- breakdown when GMRES(ℓ) stagnates

AVOIDING BREAKDOWN

two reasons for breakdown
in Bi-CGSTAB methods

(1) Bi-CG part may break-down:
Look-ahead techniques (complicated)

(2) GMRES-part gives no reduction
no expansion of Krylov subspace
solution: switch to FOM
Sleijpen and VDV '95:
Combination of GMRES and FOM

REDUCTION GMRES-ITERATIONS

restrict number of GMRES steps
by combination with other method

1. further reduction of r_i
   with e.g. GMRES itself: GMRESR
   (VDV and Vuik '94)

2. precond of search direction
   FGMRES (Saad '93)

GMRESR: more memory
robuust; reductions add

FGMRES: cheaper in memory
not robust

GMRESR

$r_0 = b - Ax_0$;

for $i = 0, 1, 2, \ldots$

  z_m approx. sol. of $Az = r_i$

   (with method of choice)

  $c = Az_m$

  for $k = 0, 1, \ldots, i-1$

    $\alpha = (c_k, c)$

    $c = c - \alpha c_k$

    $z_m = z_m - \alpha u_k$

  $c_i = c/\|c\|_2$; $u_i = z_m/\|c\|_2$

  $x_{i+1} = x_i + (c_i, r_i)u_i$

  $r_{i+1} = r_i - (c_i, r_i)c_i$

(FGMRES works with u_i)

EXAMPLE OF GMRESR

discretized Navier-Stokes problem

	GMRES(50)	GMRES	GMRESR + GMRES(10) as "solve"
MV	1220	184	198
axpy	35000	17000	1386
dot	35000	17000	1224
mem.	50	184	46
cpu	17.0	7.2	1.2

PROMISING

BREAKDOWNS IN GMRESR?

Properties for

$$\text{GMRESR} \;+ \left\{ \begin{array}{c} \text{GMRES(m)} \;+\; \text{zero start} \\ \text{or LSQR if } z_m = 0 \end{array} \right\}$$

- NO BREAK-DOWN

- FINITE METHOD

- $r_{k+1} = (I - P_k) r_0$
  P_k orth. proj. on $\{c_0, \ldots, c_k\}$

- $\|r_k\|_2 \geq \|r_{km}^{\text{GMRES}}\|_2$

- if $\|A z_m - r_k\|_2 \leq \varepsilon \quad \Rightarrow \quad \|r_{k+1}\|_2 \leq \varepsilon$

- GMRESR is scaling invariant

EXAMPLE GMRESR (2)

$$-u_{xx} - u_{yy} + \beta(u_x + u_y) = f$$

over unit square, five point stencil
$h_x = h_y = \frac{1}{100}$, $\quad \varepsilon = 10^{-12}$

	it.	matvec	cpu-time
GMRES(32)	1355	1355	69.0
CGS	288	576	5.8
Bi-CGSTAB	252	504	4.7
GMRESR(10)	36	360	12.0

$\beta = 1$

	it.	matvec	cpu-time
GMRES(4)	256	256	4.8
CGS	$n.c.$		
Bi-CGSTAB	210	420	4.6
GMRESR(10)	35	350	11.0

$\beta = 100$

EXAMPLE GMRESR (2a)

	it.	matvec	cpu-time
GMRES(4)	302	302	5.6
CGS	$n.c.$		
Bi-CGSTAB	$n.c.$		
GMRESR(10)	36	360	12.0

$$\beta = 500$$

EXAMPLE GMRESR (3)

$$-u_{xx} - u_{yy} + \beta(u_x + u_y) = f$$

over unit square, five point stencil
$h_x = h_y = \frac{1}{100}, \quad \varepsilon = 10^{-12}$
with

$$\beta = \begin{cases} 1 & \text{for} \quad x, y \in [\frac{1}{2}, \frac{3}{5}]^2 \\ 1000 & \text{elsewhere} \end{cases}$$

	it.	matvec	cpu-time
GMRES(32)	1418	1418	70.0
CGS	$n.c.$		
Bi-CGSTAB	$n.c.$		
GMRESR(10)	56	560	19.0

FOR CGS: $\|r_{583}\|_2 \leq 10^{-10}$
BUT $\|b - Ax_{583}\|_2 \approx 10^{-4}$

CHOICE OF m IN GMRESR

incompressible Navier-Stokes, flow through curved channel ($n = 1000$)

GMRES: 177 it., GMRES(50): 1323 it

m	it	m*it	cpu-time	memory
4	45	180	1.15	94
8	23	184	0.89	54
12	16	192	0.90	44
16	12	192	1.05	40
20	10	200	1.23	40

GMRESR(m)

GMRESR + LSQR(1)

effects of LSQR in phases of stagnation

$$-u_{xx} - u_{yy} + ((au)_x + au_x)/2 = 1$$

with

$$a = 20e^{3.5(x^2+y^2)}, \qquad h_x = h_y = \frac{1}{200}$$

QMR	$\|r_0\| \approx 70.0$
	$\|r_{12}\| \approx 10.0$
	$\|r_{200}\| \approx 10.0$
GMRESR +	$\|r_0\| \approx 70.0$
GMRES(8)	$\|r_2\| \approx 10.0$
	$\|r_{25}\| \approx 10.0$
GMRESR +	$\|r_0\| \approx 70.0$
GMRES(8) +	$\|r_2\| \approx 7.0$
LSQR(1)	$\|r_{25}\| \approx .12$

GCRO

GMRESR-part:
z_m approx. sol. of $Az = r_i$
(with method of choice)

When iteration method,
then sometimes effective to
keep iterations orthogonal
on previous GMRESR c_i's

Do inner-iterations with

$$A_i \equiv \prod_{j=0}^{i-1} (I - c_j c_j^T) A$$

GCRO (de Sturler and Fokkema '92)
overhead: i of innerproducts

PERFORMANCE ISSUES

Direct solution of $Ax = b$ of order n:

GAUSSIAN ELIMINATION: $\mathcal{O}(n^3)$ operations

$\Rightarrow$ n operations per entry

LAPACK: based on matrix–matrix operations

QUESTION: is the "LINPACK" benchmark representative for large sparse systems?

DENSE MATRICES

Distributed memory systems:

machine	nproc	eff.	n
CM-5	1024	45%	52,224
Intel Delta	512	70%	25,000
nCUBE 2	1024	80%	21,376
Fuj. AP1000	512	80%	25,600
MEIKO CS	62	50%	8,500

Shared memory systems:

machine	nproc	eff.	n
NEC SX-3	4	90%	6,400
CRAY C-90	16	93%	10,000

Source: Dongarra '93

SPARSE MATRICES - ICCG

Matrix order $1,000,000$
("7-diagonal")

Performance in MFLOPS for 1 processor:

machine	vect ICCG	percent.
NEC SX-3	607	22%
CRAY C-90	444	46%
IBM 9000	40	9%
CONVEX C3210	16	32%
Alliant FX2800	2.3	6%

Source: Dongarra and VDV '92

PARALLELISM

further degradation on parallel computers?

<u>parallelism at iteration level:</u>

Krylov subspace methods:

> *orthogonal basis for search space*
> *project $Ax = b$ onto search space*

v_1, ..., v_{i-1} orthogonal
next iteration step:

	$t = Av_{i-1}$
sync points	$t = t - (t, v_1)v_1 - (t, v_2)v_2, \ldots$
	$v_i = t/\|t\|$

SOME IDEAS FOR CG

* Gear & Chronopoulos ('89): combine 2 steps

* Meurant ('87): reduce 2 sync points to 1

* d'Azevedo et al ('92), Eijkhout ('92): rewrite some innerproducts

main idea: exploit intrinsic orthogonality properties: $(r_i, r_j) = 0$ and $(p_i, Ap_j) = 0$

CONJUGATE GRADIENTS (1)

$r_0 = b - Ax_0$; $p_{-1} = 0$; $\beta_{-1} = 0$
Solve $Kw_0 = r_0$; $\rho_0 = (r_0, w_0)$
for $i = 0, 1, 2, \ldots$

$p_i = w_i + \beta_{i-1} p_{i-1}$ (*update*)
$q_i = Ap_i$
$\alpha_i = \rho_i / (p_i, q_i)$ (*dot*)
$x_{i+1} = x_i + \alpha_i p_i$ (*update*)
$r_{i+1} = r_i - \alpha_i q_i$ (*update*)
Solve $Kw_{i+1} = r_{i+1}$
$\rho_{i+1} = (r_{i+1}, w_{i+1})$; (*dot*)
$\beta_i = \rho_{i+1} / \rho_i$

COMPUTATIONAL INGREDIENTS

MATRIX $\times$ vector: Ax
and preconditioning

$\mathcal{O}(n)$ ops / $\mathcal{O}(n)$ data

innerproducts: $\sum_{i=1}^{n} x_i y_i$

$\mathcal{O}(n)$ ops / $\mathcal{O}(n)$ data

vector updates: $x = y + \alpha * z$

$\mathcal{O}(n)$ ops / $\mathcal{O}(n)$ data

COMMUNICATION AND CPU-TIME

CPU- and Comm. time per CG-iteration:

nprocs	CPU-time	comm.tijd	total
1	0.1700	0.0000	0.1700
2	0.0850	0.0103	0.0950
4	0.0425	0.0118	0.0540
8	0.0210	0.0133	0.0350
16	0.0106	0.0148	0.0250
32	0.0053	0.0163	0.0220
64	0.0027	0.0178	0.0205
128	0.0013	0.0193	0.0206

Source: Crone '93 (for Meiko CS)

CG (2) (Chrono. & Gear)

for $i = 0, 1, 2, \ldots$

$$
\begin{aligned}
& p_i = w_i + \beta_{i-1} p_{i-1} \\
& q_i = s_i + \beta_{i-1} q_{i-1} \qquad (q_i = A p_i) \\
& x_{i+1} = x_i + \alpha_i p_i \\
& r_{i+1} = r_i - \alpha_i q_i \\
& \text{Solve } K w_{i+1} = r_{i+1} \\
& s_{i+1} = A w_{i+1} \\
& \rho_{i+1} = (r_{i+1}, w_{i+1}); \\
& \mu_{i+1} = (s_{i+1}, w_{i+1}) \\
& \beta_i = \rho_{i+1}/\rho_i; \ \alpha_{i+1} = \frac{\rho_{i+1}}{\mu_{i+1} - \rho_{i+1}\beta_i/\alpha_i}
\end{aligned}
$$

only 1 synchronization point

numerical stability?

CG (3) (D, H, & VdV) $K = LL^T$

for $i = 0, 1, 2, \ldots$

$$w_i = L^{-T} s$$
$$p_i = w_i + \beta_{i-1} p_{i-1}$$
$$q_i = A p_i;\ \gamma = (p_i, q_i)$$
$$x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$$
$$\alpha_i = \rho_i / \gamma$$
$$r_{i+1} = r_i - \alpha_i q_i$$
$$s = L^{-1} r_{i+1};\ \rho_{i+1} = (s, s)$$
$$\beta_i = \rho_{i+1} / \rho_i$$

no synchronization point

overlap

Bi-CGSTAB(2) Part A

$r_0 = b - Ax_0$; choose $\widehat{r}$

further initializations

for $i = 0, 2, 4, \ldots$

$$\left.\boxed{\begin{array}{l}\rho = \widehat{r}^T r_i \\ u = u - \omega_1 v - \omega_2 w\end{array}}\right\} \textit{overlap}$$

$$\beta = \cdots; \;\cdots\cdots\; v = Au$$

$$\left.\boxed{\begin{array}{l}\gamma = \widehat{r}^T v \\ u = r_i - \beta u\end{array}}\right\} \textit{overlap}$$

$$\alpha = \cdots; \; r = r_i - \alpha v; \; s = Ar;$$

$$\left.\boxed{\begin{array}{l}\rho = \widehat{r}^T s \\ x = x_i + \alpha u\end{array}}\right\} \textit{overlap}$$

$$\beta = \cdots; \; v = s - \beta v; \; w = Av$$

$$\left.\boxed{\begin{array}{l}\gamma = \widehat{r}^T w \\ u = r - \beta u\end{array}}\right\} \textit{overlap}$$

$$\alpha = \cdots; \; r = r_i - \alpha v; \; s = s - \alpha w; \; t = As;$$

Bi-CGSTAB(2) Part B

$$\left.\boxed{\begin{array}{l}\xi_1 = r^T s;\ \mu = s^T s \\ \nu = s^T t;\ \tau = t^T t \\ \xi_2 = r^T t\end{array}}\right\}\ \textit{combine}$$

$x = x + \alpha u$

$\omega_1 = \cdots;\ \omega_2 = \cdots$

$\cdots\cdots\cdots\cdot$

$x_{i+2} = x + \omega_1 r + \omega_2 s$

$r_{i+2} = r - \omega_1 v - \omega_2 w$

end of for-loop

PARALLEL ASPECTS OF GMRES

v_i: orthogonal basis for $K^m(A; r_0)$

PROBLEM: m DOTS (sequential process)

ALTERNATIVE: first compute a basis

$$\{v_1, p_1(A)v_1, \ldots, p_{m-1}(A)v_1\}$$

then orthogonalize $\{y_i = p_i(A)v_1\}$

possibilities for parallelism

PARALLEL ORTHOGONALIZATION

De Sturler '91

FOR $i = 2, ..., m$

    FOR $j = i, ..., m$

        $y_j = y_j - (y_j, v_{i-1})v_{i-1}$ (parallel)

    $v_i = y_i / \|y_i\|_2$

combine messages for DOT's
less start-up time

OVERLAP IN ORTH.

FOR $j = i, ..., m$

$$y_j = y_j - (y_j, v_{i-1})v_{i-1}$$

1. split loop in two parts
2. compute local DOT's group 1
3. assemble local DOT's: communication
4. compute local DOT's group 2
5. update y_i; local DOT's for $\|y_i\|$
6. assemble DOT's group 2: communication
7. update $y_i, ..., y_m$
8. compute $v_i = y_i/\|y_i\|_2$

PRECONDITIONING

Given $Ax = b$, find $M = LU \approx A$
and solve $M^{-1}Ax = M^{-1}b$

$M^{-1}A$ only in matrix vector op's:

$$q = M^{-1}Ap$$

1. $\tilde{p} = Ap$

2. Solve q from $Mq = \tilde{p}$

That is:

2a $Lz = \tilde{p}$

2b $Uq = z$

PROBLEMS WITH PARALLELISM

APPROACHES FOR PARALLELISM

- parallelism without changing num. properties
- modified algorithms

Iterative methods: globality
Parallel methods: locality

Four approaches:
1. reordering
2. series expansion
3. domain decomposition
4. explicit approx. inverses

REORDERING OF UNKNOWNS

1. Wavefront ordering:
   exploit dependency graph

2. Red-Black Ordering:
   RED points in parallel
   BLACK points also

Duff & Meurant, '89: EFFECTS

Kuo & Chan, '90: CG + R-B-ILU:
about twice as fast as CG

DeLong & Ortega, '94:
a. SOR works well for Red-Black
b. k steps SOR as preconditioner
c. experiments: very effective
d. costs: for k steps SOR k MV's

<u>highly parallel</u>

SOME REFERENCES

* Freund, Golub, and Nachtigal
  Acta Numerica, 1992

* Dongarra, Duff, Sorensen, and
  Van der Vorst, SIAM '91

* Pommerell: Solution of large unsymmetric
  systems, Hartung–Gorre, 1992

* Barrett, Berry, Chan, Demmel, etc
  Templates, SIAM 1994,
netlib, WWW

* Home page Henk van der Vorst
URL: http://www.math.ruu.nl/people/vorst